SELF-OPTIMIZING ARCHITECTURE IN MOBILE MACHINES

Timo KAUTZMANN[1] and Marcus GEIMER[2];
Micaela WÜNSCHE[1] and Hartmut SCHMECK[2]
Karlsruhe Institute of Technology (KIT)
Institute of Mobile Machines (MOBIMA) and Institute of Applied Informatics and Formal Description Methods (AIFB)
Kaiserstraße 12
76131 Karlsruhe
Germany
Email: timo.kautzmann@kit.edu

Today's machine management systems in off-highway machines are designed to optimize with respect to a target function without integrating the entire machine or considering environmental interactions. For that reason the interdisciplinary project OCOM – "Organic Computing in Off-highway Machines" started in February 2009 to design an architecture for an off-highway machine in order to close that gap. Optimization of fuel consumption is exemplarily chosen even though many other goals are reachable. This paper will introduce the generic architecture; first results will be presented.

Keywords: Machine Management, Organic Computing, Self-learning, Generic Optimization Architecture, Fuel Consumption

1 INTRODUCTION

New developments in the field of hydraulics and drive train technology in off-highway machines lead to a steady increase of degrees of freedom. The human operator is not able to set all of them, that's why an automated overall machine management (OMM) is necessary to relieve him. An OMM is the combination of Hard- and Software to realize an operating strategy. The architecture of today's OMM is illustrated in Figure 1. The operator sets the basic defaults which are input into static characteristic curves or arrays to find in predefined cases optimized command variables for the single subsystems. Command variables of one subsystem are basically set without respect to settings of other subsystems. In each subsystem, command variables are controlled individually without considering interactions between other subsystems. The output of each subsystem is accumulated to a target working result that is measured and controlled by the operator. Subsection 2.1 will give some examples.
As a result, current management systems aren't able to fulfil holistic optimization. Holistic optimization will be understood as follows:

- Holistic optimization is supposed to consider environmental influences like attributes set by the operator or the current working cycle. Considering that the settings of the operator are in some cases sub-optimal, an automated optimization with respect to the input of the driver is necessary. Furthermore, off-highway machines, for instance tractors, perform a tremendous

[1] PhD candidate
[2] Supervisor

number of different working cycles. The optimal parameter settings for transportation are for example not optimal for heavy duty plowing. Due to the fact that today's automated machine management system cannot recognize these different situations, it has to find compromises according to defaults of an implemented target function while setting its working point.

- Holistic optimization is supposed to regard the system as a whole. Since a mobile machine is a deeply cross-linked system, holistic optimization must aggregate all possible influences and know how they may interact with each other. Currently, this is not possible, as each subsystem considers its local parameters only, because of the lack of an entire machine observation and controlling.

In this paper a new controlling approach with self-adaptive and learning capabilities will be presented that is able to recognize the environment and perform holistic optimization according to the definition above.

The architecture will be introduced in Subsection 2.2 before Section 3 will outline the main components. Due to the fact that there is a tremendous number of different influences to an off-highway machine, an a priori adaptation of parameters to optimized values is not possible. Hence, the architecture needs to be equipped with certain learning capabilities to find best settings in current situations.

Final goal is to integrate the architecture into a test vehicle. Section 4 describes the intermediate steps to reach that goal. Section 5 closes this paper with a conclusion.

Even though the testing vehicle is a Fendt Vario 412 from AGCO GmbH, the intention is to systematically generalize the architecture to other off-highway machines. The currently considered goal is to optimize fuel consumption; however, additional goals are easily conceivable. Preceding publications in this area are **Bliesener, M.** (2009) and **Kautzmann, T.; Wünsche, M. et al** (2010).

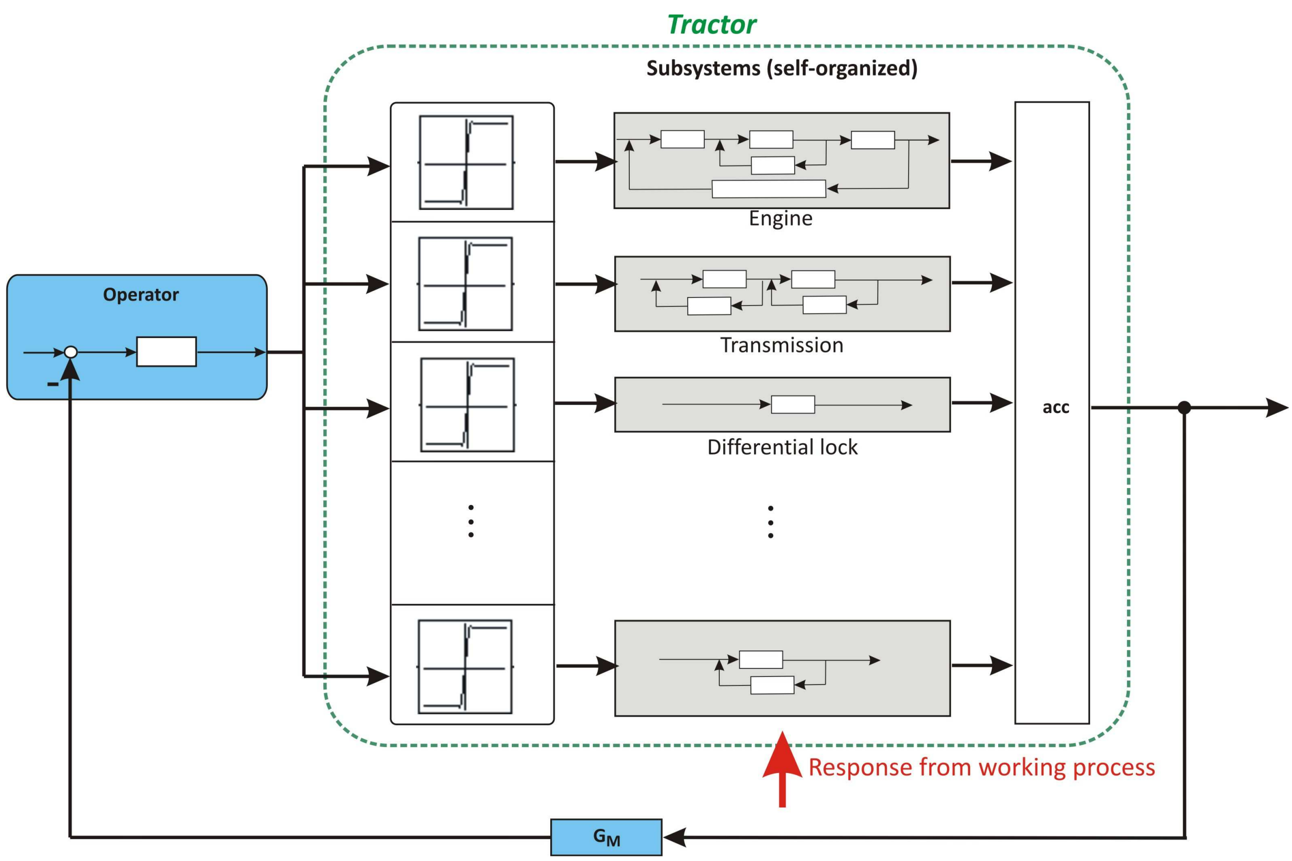

Fig.1: Overall Machine Management

2 RELATED WORK

2.1 Today's Management Systems

A tractor is both a self-organized and cross-linked system as we will see hereafter. Today's management systems however optimize only parts of this cross-linked system. Reduction of fuel consumption for example is basically achieved by keeping efficiency at a maximum with reduced shaft speed [**Brunotte, D.** (2001), **Seeger, J.** (1999), **Haas, W.** (2002)]. AGCO's "Tractor Management System" (TMS) controls engine and transmission while the driver sets desired tractor speed. Engine speed rises according to imprinted load. Furthermore the system switches off four-wheel drive while driving above a certain velocity to eliminate circulating power in the drive train according to Reiter's suggestion [**Reiter, H.** (1990)]. Tractor management in a Fendt Vario also deactivates the differential lock at a certain steering angle to avoid bad traction conditions between wheel and soil. Other systems also integrate pivoting angle and pressure of the hydraulic pump in order to control engine overloading [**Forche, J.** (2003)].
However not yet realized in production vehicles, one first attempt to combine a greater amount of sub-systems of a tractor is done by **Jaufmann, A.** (1997). He links the management systems of the transmission, engine, chassis, and lifting device in order to reduce fuel consumption. According to Jaufmann a "distinct" reduction of fuel consumption is achievable depending on the working cycle. **Frerichs, L.** (1991) combines tractor and accessory equipment based on tractor-plow aggregate. Control variables are traction force and slip. Actuating variables are working depth and width, location of plow according to tractor, gear ratio of the transmission and shaft speed. A "distinct" optimization of performance and fuel consumption is reachable, too. **Kipp, C.** (1987) implemented the mentioned management based on digital controlling systems and microcomputers and reached an optimization of 15 % both in performance and fuel consumption.

2.2 Alternative Control Architectures

Today's management of an off-highway machine, as seen in Subsection 2.1, can be characterized more generally. The system performs certain working cycles. The associated tasks can only be managed adequately if all individual subsystems work and especially cooperate. In other words, the main task can only be fulfilled by a combination of subsystems that have to coordinate their actions. Today, subsystems are handled individually. To perform the tractor working cycle "disc-harrowing", for instance, subsystems like combustion engine, transmission drive, power take-off (PTO) and many more are needed in a specific order, where each subsystem has to deliver the right performance at the right time. Such a system is called self-organized. To control such systems efficiently in every way, architectures are required that are both robust and flexible at the same time.
In literature, several approaches have been introduced. An approach that especially emphasizes a self-learning ability of the system, offline as well as online, in order to adjust it even to previously unknown situations is that of the Observer/Controller-architecture (O/C) of the Organic Computing Initiative [**Schmeck, H.** (2005), **Branke, J.; Mnif, M. et al** (2006), Richter, **U.; Mnif, M. et al** (2006), **Mnif, M.; Richter, U. et al** (2007)].
Here, the system under consideration (e.g. the tractor) is called a System under Observation and Control (SuOC). This SuOC is capable of performing its intended function on its own, without interference, but not necessarily in an optimal way. The O/C-architecture is intended to optimize the performance and supervise the system as a whole consisting of independent but cooperating

subsystems. At the same time it provides an interface for a system user (or a higher level entity) to provide specific optimization objectives (see Figure 2).
To do this, an Observer records and analyzes at all times the status of the SuOC, and reports an aggregated description of the current status to a Controller. This Controller decides whether the system status requires an action, and if so, takes it to influence system performance.
High-level optimization objectives can be given to the Controller. Depending on the observed situation, the controller can also influence settings in the Observer by switching between different models of observation.

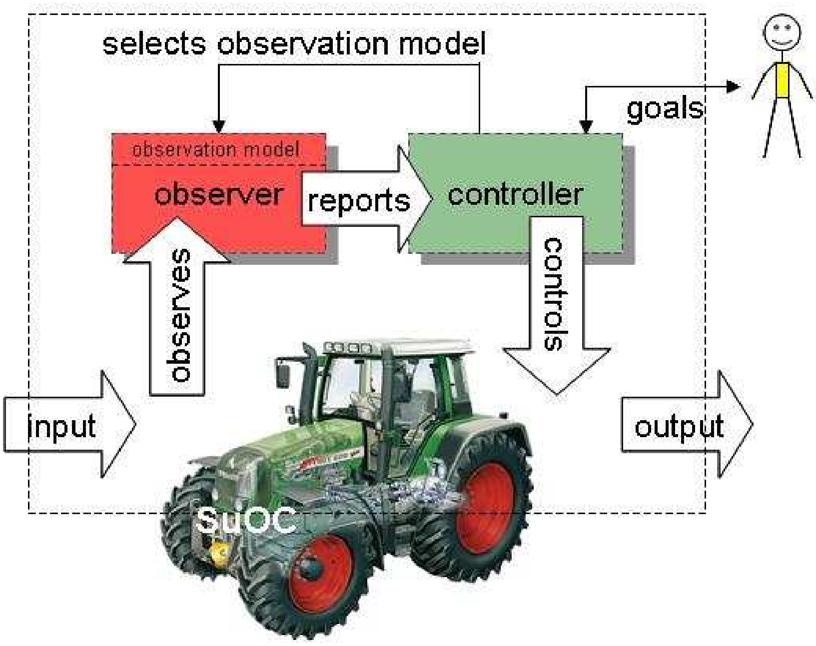

Fig. 2: Observer/Controller Architecture [**Bliesener, M.** 2009]

3 ARCHITECTURE

3.1 System under Observation and Control

The system we want to observe and control is a Fendt Vario 412. Figure 3 gives an overview over the infinite variable hydraulic-mechanical transmission ML90 in a Fendt Vario 412 which provides maximum degrees of freedom.
Combustion engine is the power source for the main power flows in a tractor: traction, PTO and hydraulic power. A change in one power flow changes the whole system state and thereby fuel consumption. Therefore a tractor is a deeply cross-linked entity; alterations in one part may lead to a completely new system state.

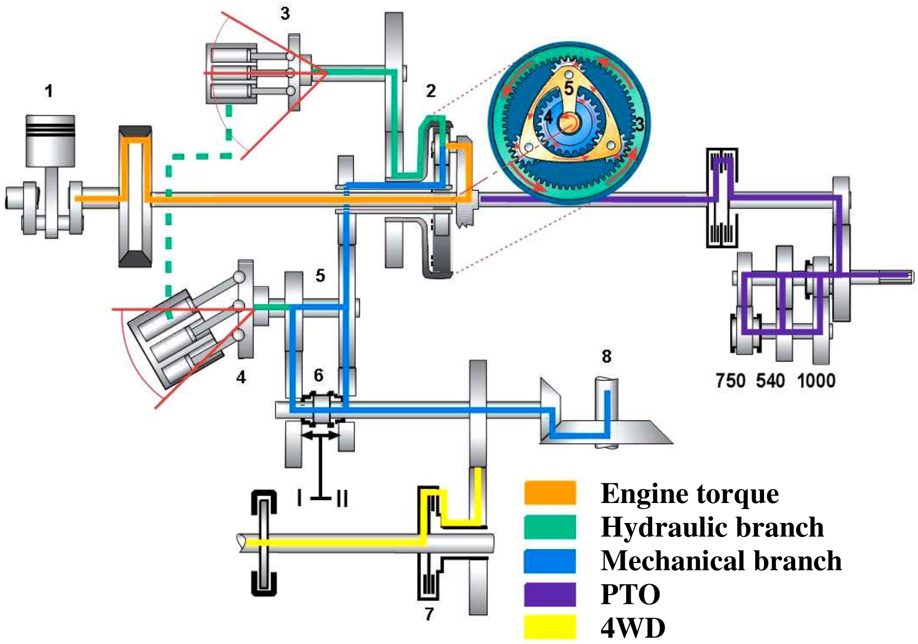

Fig. 3: Transmission scheme of a Fendt Vario; © AGCO GmbH

Sensor signals to the observer ("raw data") are shown in Figure 4 as well as actuator signals from the controller ("action").

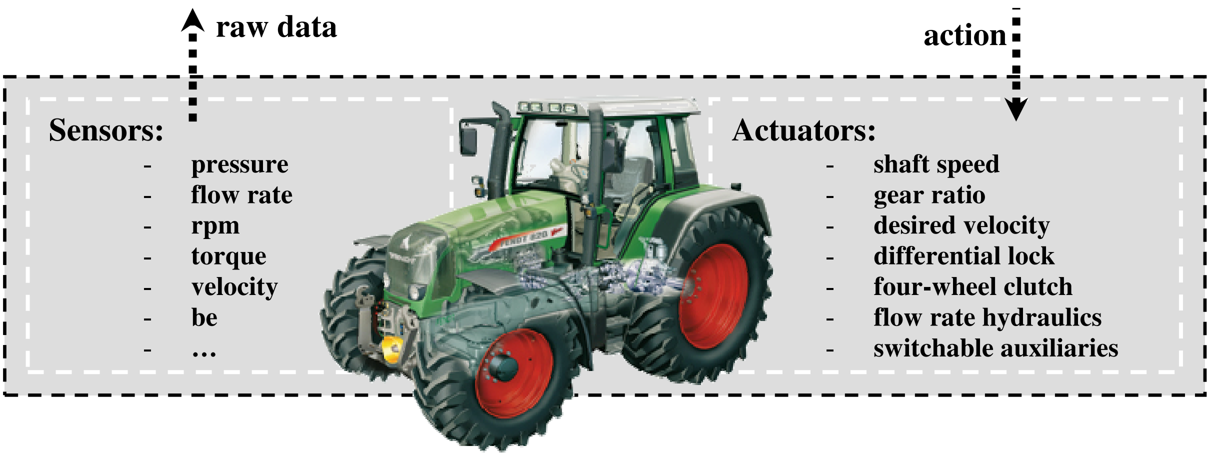

Fig. 4: System under Observation and Control [**Kautzmann, T.** 2010]

3.2 Observer

The Observer part of the described O/C-architecture continually monitors the System under Observation and Control (SuOC). An internal schematic is shown in Figure 5. Sensor data is sampled by the *monitor* module, both concerning overall system status and individual data from specific components of the machine. Which sensor data is to be read, and at which sampling frequency, is specified in the observation model set by the Controller part of the O/C-architecture. All sampled data is then stored in the *log* module, for possible later use.
In the *pre-processor* module, the monitored data is cleared from noise and outliers by low-pass filtering, before it is evaluated in the *data analyzer* module.
In data analysis, statistical values are derived from specified time windows over the incoming data stream, like arithmetic mean or minimum and maximum value. Also, linear regression and clustering of data points are performed, in order to identify inherent patterns. The aim is to identify the current working cycle of the machine, in order to enable the Controller to adjust all components of the machine appropriately.
To further help the Controller in this task, the *predictor* module of the Observer also receives the system state from the *data analyzer*, and on the basis of this data and the experience it accumulates over time, the following system state is predicted.
All the information gathered within the modules of the Observer is then collected by the *aggregator* module, and passed on to the Controller part of the O/C-architecture.

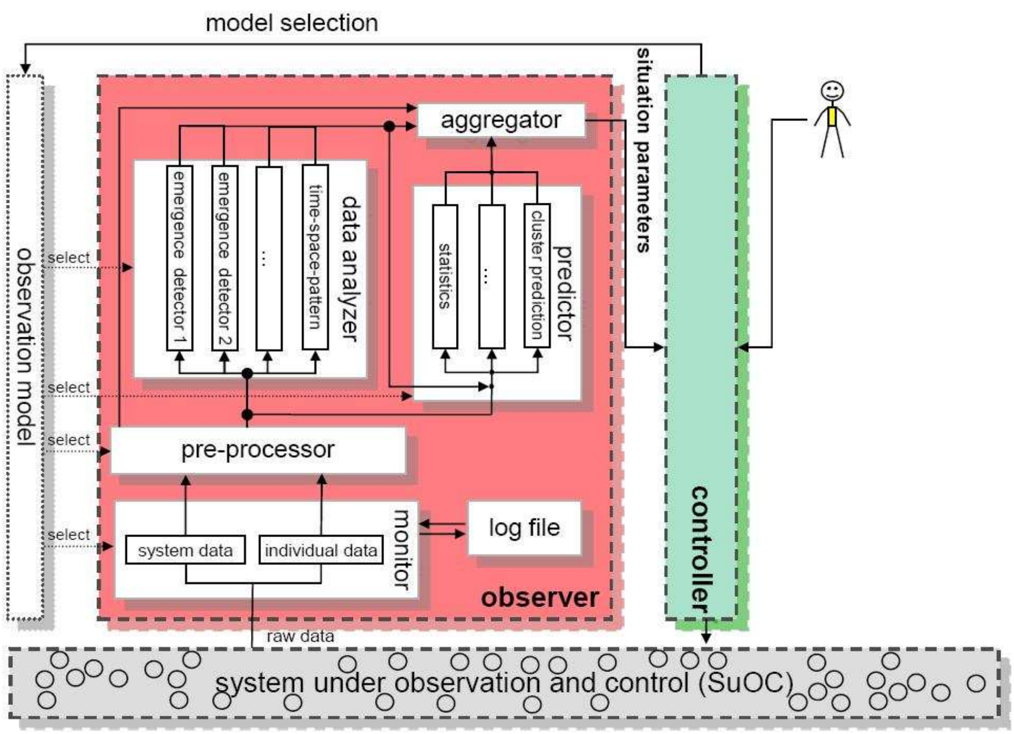

Fig. 5: Observer [**Richter, U.; Mnif, M. et al** (2006)]

3.3 Controller

The Controller part of the O/C-architecture receives all relevant information about the system state from the Observer, and on this basis decides in which way to influence the machine. Figure 4 shows

the different possibilities for the Controller to act. Internal schematics of the Controller are shown in Figure 6.
For selection of a specific action, the Controller has a *mapping* module that assigns to a system state C_i an appropriate action A_i. Over time, the Controller will adjust this mapping, thus learning to apply the best action in every situation, even if the situation was previously unknown. This learning process takes place at two levels, *online* and *offline*, as is explained in the following.
If the currently reported system state is already part of the mapping, the according action is taken and then stored in the *action history*. After some time steps t, the system state is again recorded in the *situation parameters history*, and considered the outcome of action A_i in situation C_i. Depending on whether the outcome was positive or negative, the corresponding situation-action mapping is evaluated and assigned a fitness value. In this way, the system learns *online* which mappings are best suited. Eventually, mappings with a low fitness will be replaced.
However, if the currently reported system state is not part of the mapping, no immediate action is taken by the Controller to influence the machine. Instead, a *simulation model* of the machine, that is part of the Controller, is initialized with the system state C_i. An *adaption* module generates new rules, tests them *offline* in simulation and evaluates the simulated outcome. The best new rule is then incorporated into the mapping. The same method of *offline learning* of new rules is also used when replacing mappings with a low fitness.
The basis for the evaluation of situation-action mappings always consists in objectives that are provided externally. In this case, it is a reduction of fuel consumption, but the optimization process can be influenced in any direction by imposing an alternative goal.

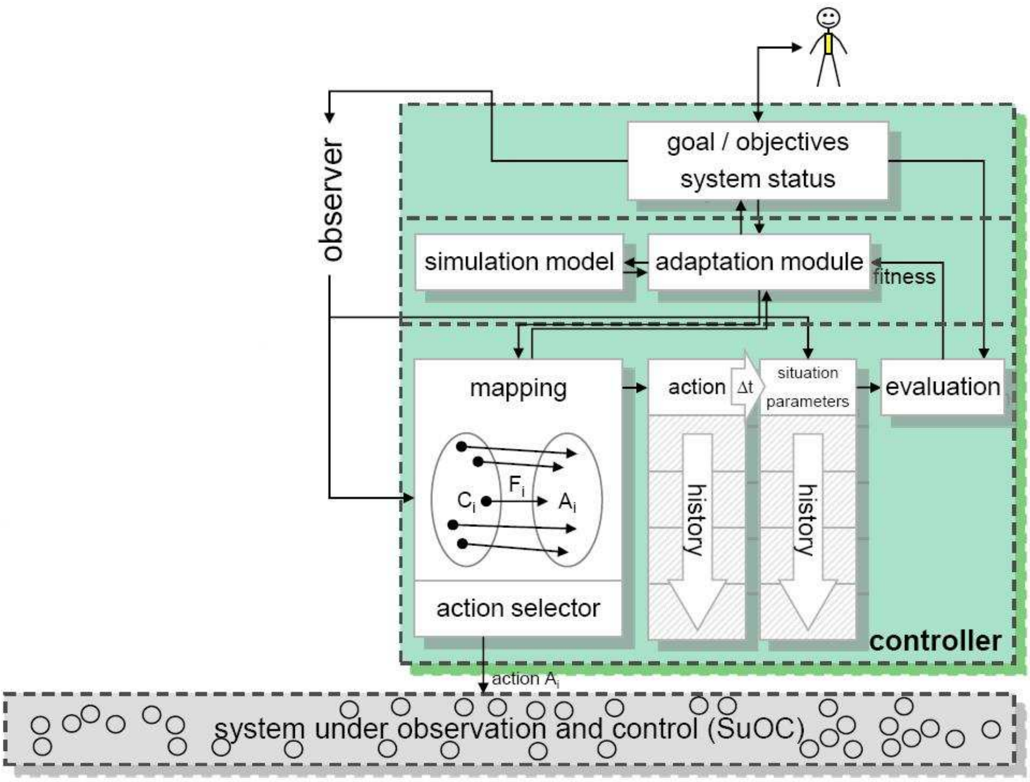

Fig. 6: Controller [**Richter, U.; Mnif, M. et al** (2006)]

4 PRESENT ISSUES

Present work focuses on creating quasi-static and efficiency- afflicted simulation models both in *MATLAB/ Simulink* and *AMESim*. *AMESim* is a topology- orientated simulation program of *LMS*

Imagine.Lab. *MATLAB/ Simulink* was chosen to be the platform of the O/C architecture. Therefore the *Simulink- model* serves the controller to learn offline as described in Subsection 3.3. The *AMESim- model* is meant to simulate the SuOC in the first step to both provide all needed sensor signals for the Observer and simplifies the realization of *action* signals from the Controller. As input for the *AMESim-* model *PowerMix* cycles of the *German Agricultural Society (DLG)* are used since they describe the main working processes a tractor performs. *PowerMix* provides traction, PTO and hydraulic power over time. Communication between *AMESim* and *Simulink* is realized via so-called S-function. Results of that "MiL" (Model in the Loop) – simulation is the validation of the architecture and the development of a requirements list for the communication between the real tractor and the O/C architecture.
The *Simulink-* model is designed as shown in Figure 7. According to Figure 6 the module *simulation model* receives all information from *adaptation module*. The model calculates the power flow from required output at the implement to necessary power at the engine. Therefore the *implement model* in Figure 7 computes the power flow at the interface between implement and tractor as a result of needed traction, PTO and hydraulic power depending on current velocity. In *Drive train*, *PTO* and *Working hydraulics model* power flows are transmitted and added to total required rotational power at the crank shaft and finally to fuel consumption of *combustion engine*. The results are transferred again to the *adaptation module*.
To design efficiency- afflicted models a time-based simulation approach with a physical loss modelling and concentrated parameters is chosen. To determine efficiencies of the hydrostatic entities, 4D characteristics are used.

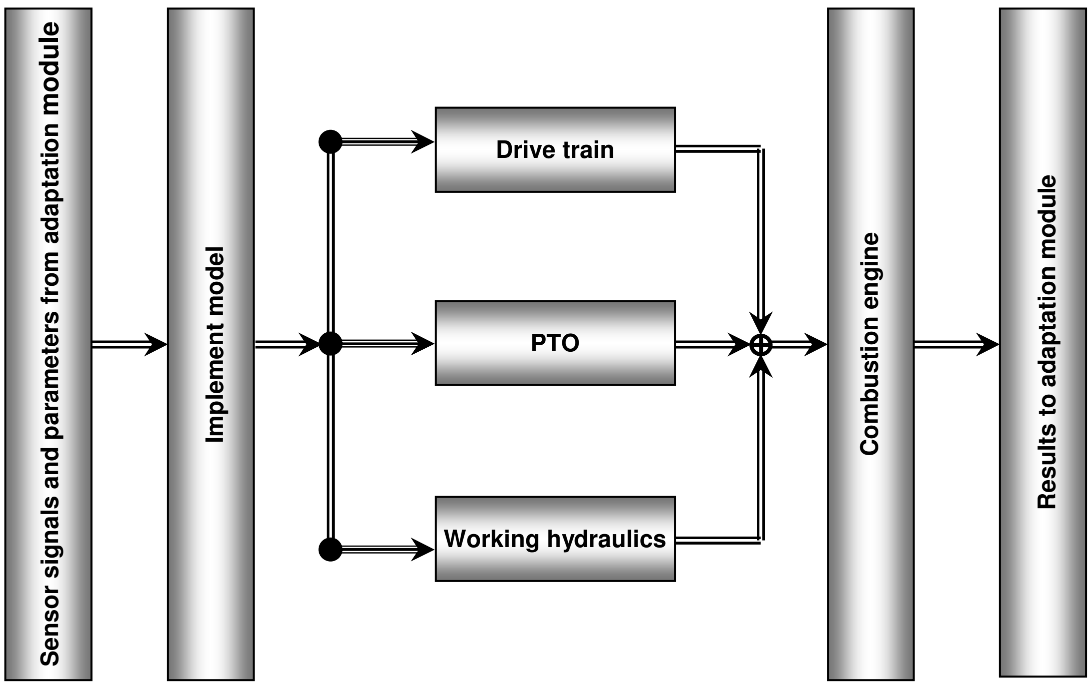

Fig. 7: Design of the Matlab/ Simulink simulation model

5 CONCLUSION

In this paper we outlined an architecture for a self-learning machine management system of an off-highway machine that is based on the generic O/C architecture. The management will be adapted to

an existing machine and will be able to perform holistic optimization as described in Section 1. Since additionally equipped with self-learning capabilities, it will be able to evolve permanently both under unforeseen conditions and changing environment. As shown at the Agritechnica 2009 in Hannover (Germany), comfort is a mayor issue. In this context the architecture is able to improve comfort by relieving the driver.
In OCOM, fuel consumption is exemplarily chosen for optimization. **Schreiber, M**. (2006) examines the potentials of an overall machine optimization that includes the environment and regards the system as a whole like the described O/C architecture. According to him, an average of 5 to 25 % fuel consumption reduction compared to existing machine management systems and for certain working cycles up to 30 % are realistic.

6 ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support by the German Research Foundation (DFG). Furthermore we want to thank AGCO Fendt for their support and helpful hints.

7 LIST OF ABBREVIATIONS

4WD	Four-wheel drive
be	Fuel consumption
DFG	German Research Foundation
DLG	German Agricultural Society
MIL	Model in the Loop
O/C	Observer/ Controller
OCOM	Organic Computing in Off-highway Machines
OMM	Overall Machine Management
PTO	Power Take Off
rpm	Revolutions per minute
SuOC	System under Observation and Control
TMS	Tractor Management System

8 REFERENCES

Bliesener, M. (2009). Organic Computing in Offhighway Machines. *In Proceedings of Mobile Machines – Sicherheit und Fahrerassistenz für Arbeitsmaschinen 2009, Vol. 1, n.n***.**

Branke, J.; Mnif, M.; Müller-Schloer, C.; Prothmann, H.; Richter, U.; Rochner, F.; Schmeck, H.(2006). Organic Computing – Addressing complexity by controlled selforganization. *In Proceedings of the 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006).*

Brunotte, D. (2001). Traktormanagement auf dem Systemfahrzeug XERION im Feldversuch. *Landtechnik 2001, pp. 35-40.*

Forche, J. (2003). Management hydraulischer Antriebe in mobilen Arbeitsmaschinen. *Landtechnik 2003, pp. 239-244.*

Frerichs, L. (1991). *Grundlagen für das elektronische Traktor-Pflug-Management*. Forschungsbericht der Agrartechnik der MEG Nr. 204, Hohenheim.

Haas, W. (2002). Der Tempomatregler als Beispiel für ein modernes Antriebsstrangmanagement. *Landtechnik 2002, pp. 31-36.*

Jaufmann, A. (1997). Untersuchung eines Traktormanagements. *Landtechnik 1997, pp. 17-20.*

Kautzmann, T.; Wünsche, M.; Geimer, M.; Schmeck, H.; Mostaghim, S. (2010). Self-optimizing Machine Management. *In Proceedings of 2nd International Conference on Machine Control and Guidance, Institut für Geodäsie und Geoinformation. Schriftenreihe 16. pp. 107-115.*

Kipp, C. (1987). *Optimierung des Leistungsumsatzes von Traktoren durch Einsatz elektronischer Hilfsmittel*. Dissertation Technische Universität Berlin, Berlin.

Mnif, M.; Richter, U.; Branke, J.; Schmeck, H.; Müller-Schloer, C. (2007). Measurement and control of self-organised behaviour in robot swarms. *In P. Lukowicz, L. Thiele, and G. Tröster, editors, Proceedings of the 20th International Conference on Architecture of Computing Systems (ARCS 2007), , Vol 4415 of LNCS, pp. 209–223.*

Reiter, H. (1990). *Verluste und Wirkungsgrade in Traktorgetrieben*. VDI, Düsseldorf.

Richter, U.; Mnif, M.; Branke, J.; Müller-Schloer, C.; Schmeck, H.(2006). Towards a generic observer/controller architecture for Organic Computing. *In C. Hochberger and R. Liskowsky, editors, INFORMATIK 2006 – Informatik für Menschen!, Köllen, Vol. P-93 of GI-Edition – Lecture Notes in Informatics (LNI), pp. 112–119.*

Schmeck, H. (2005). Organic Computing -- A New Vision for Distributed Embedded Systems. *In Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 201--203, Los Alamitos, CA, USA.*

Schreiber, M. (2006). *Kraftstoffverbrauch beim Einsatz von Ackerschleppern im besonderen Hinblick auf CO2-Emissionen.* Shaker, Aachen.

Seeger, J. (1999). Untersuchungen von Antriebsstrangstrategien eines TMS. *Landtechnik 1999, pp. 93-98.*